

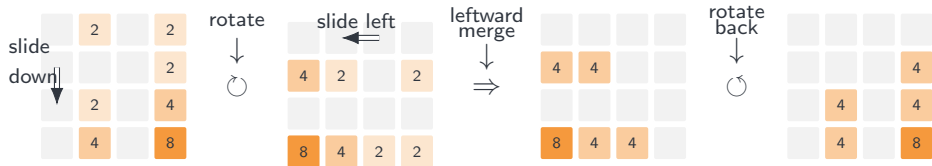
FOUNDATIONS FOR SYMMETRIC LOGIC

Ohad Kammar
University of Edinburgh

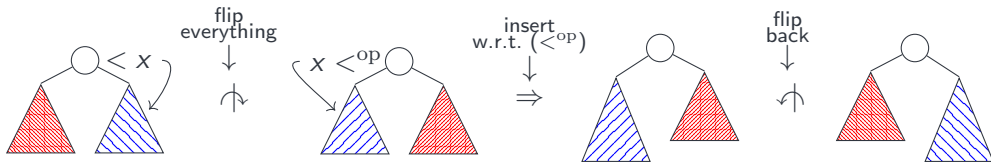
Matija Pretnar
University of Ljubljana

Previously, on **WG 2.1...**

The **easiest** way to implement 2048 is using **symmetry**



The **most interesting** way to implement tree insertion is using **symmetry**



Schur's inequality is proven using w.l.o.g.

Theorem

Take $t > 0$. Then, for all $x_1, x_2, x_3 > 0$, we have:

$$x_1^t(x_1 - x_2)(x_1 - x_3) + x_2^t(x_2 - x_3)(x_2 - x_1) + x_3^t(x_3 - x_1)(x_3 - x_2) \geq 0$$

Proof

Without loss of generality, assume $x_1 \geq x_2 \geq x_3$. Then

$$\begin{aligned} x_1^t(x_1 - x_2)(x_1 - x_3) + x_2^t(x_2 - x_3)(x_2 - x_1) + x_3^t(x_3 - x_1)(x_3 - x_2) &= \\ (x_1 - x_2) \left(x_1^t(x_1 - x_3) - x_2^t(x_2 - x_3) \right) + x_3^t(x_1 - x_3)(x_2 - x_3) &\geq \\ (x_1 - x_2)(x_1^t - x_2^t)(x_1 - x_3) + x_3^t(x_1 - x_3)(x_2 - x_3) &\geq 0 \end{aligned}$$



Schur's inequality is proven using w.l.o.g.

Theorem

Take $t > 0$. Then, for all $x_1, x_2, x_3 > 0$, we have:

$$x_1^t(x_1 - x_2)(x_1 - x_3) + x_2^t(x_2 - x_3)(x_2 - x_1) + x_3^t(x_3 - x_1)(x_3 - x_2) \geq 0$$

Proof

Without loss of generality, assume $x_1 \geq x_2 \geq x_3$. Then

$$\begin{aligned} x_1^t(x_1 - x_2)(x_1 - x_3) + x_2^t(x_2 - x_3)(x_2 - x_1) + x_3^t(x_3 - x_1)(x_3 - x_2) &= \\ (x_1 - x_2) \left(x_1^t(x_1 - x_3) - x_2^t(x_2 - x_3) \right) + x_3^t(x_1 - x_3)(x_2 - x_3) &\geq \\ (x_1 - x_2)(x_1^t - x_2^t)(x_1 - x_3) + x_3^t(x_1 - x_3)(x_2 - x_3) &\geq 0 \end{aligned}$$



How do we **mechanise** w.l.o.g.?

Rocq

The general syntax of without loss is:

```
wlog suff? clear_switch? i_item? : ident* / term
```

On a goal G , it creates two subgoals: a first one to prove the formula $(\text{term} \rightarrow G) \rightarrow G$ and a second one to prove the formula $\text{term} \rightarrow G$. If the optional list of `ident` is present on the left side of `/`, these constants are generalized in the premise $(\text{term} \rightarrow G)$ of the first subgoal.

How do we **mechanise** w.l.o.g.?

Agda

```
-- Without Loss of Generality
module _ {_R_ : Rel A ℓ1} {Q : Rel A ℓ2} where

wlog : Total _R_ → Symmetric Q →
      (∀ a b → a R b → Q a b) →
      ∀ a b → Q a b
```

How do we **mechanise** w.l.o.g.?

Lean

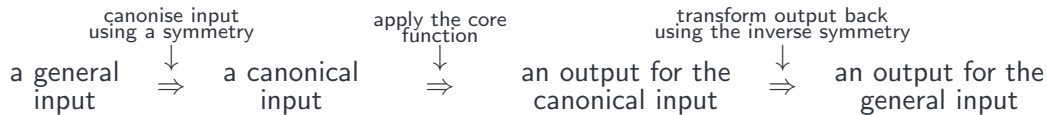
The tactic `wlog h : P` will add an assumption `h : P` to the main goal, and add a new goal that requires showing that the case $\neg h : P$ can be reduced to the case where `P` holds (typically by symmetry).

The most uniform way of using w.l.o.g. is using symmetry

$$\begin{array}{ccccccc} & \text{sort using } \pi & & \text{prove using} & & \text{permute back} & \\ & \downarrow & & \text{additional assumption} & & \downarrow & \\ & \text{since } P \text{ is invariant} & & & & & \\ x_1, x_2, x_3 \in \mathbb{R}_+ & \Rightarrow & x_{\pi 1} \geq x_{\pi 2} \geq x_{\pi 3} & \Rightarrow & P(x_{\pi 1}, x_{\pi 2}, x_{\pi 3}) & \Rightarrow & P(x_1, x_2, x_3) \end{array}$$

The general construction

Let us identify the **basic components**



To capture symmetric sets, we employ **group actions**

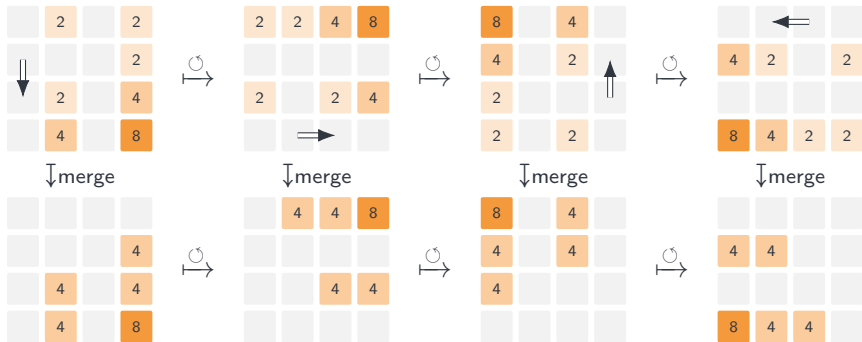
Given a group G , a G -**action** A consists of:

- a **carrier** set $\underline{A}$;
- an **action** map $(\overset{A}{*}) : \underline{G} \times \underline{A} \rightarrow \underline{A}$;

such that:

- $e * x = x$;
- $(g_1 * g_2) * x = g_1 * (g_2 * x)$.

Equivariant maps are symmetry-preserving morphisms



An **equivariant map** $\varphi : A \circledast \rightarrow B$ is a map $\varphi : \underline{A} \rightarrow \underline{B}$ such that $\varphi(g \circledast a) = g \circledast (\varphi a)$ for all $g \in \underline{G}$ and $a \in \underline{A}$.

Canonisers $c : \underline{A} \rightarrow \underline{G}$ describe how to obtain a **canonising symmetry**

$(*) : \mathbf{C}_4 \times \text{Dir} \rightarrow \text{Dir}$

$\text{Dir} := \{\Rightarrow, \Uparrow, \Leftarrow, \Downarrow\}$

$\circlearrowleft * \Rightarrow := \Uparrow \quad \circlearrowleft * \Uparrow := \Leftarrow \quad \circlearrowleft * \Leftarrow := \Downarrow \quad \circlearrowleft * \Downarrow := \Rightarrow$

$\text{mkLft} : \text{Dir} \rightarrow \underline{\mathbf{C}_4}$

$\text{mkLft } \Rightarrow := \circlearrowleft^2 \quad \text{mkLft } \Uparrow := \circlearrowleft \quad \text{mkLft } \Leftarrow := e \quad \text{mkLft } \Downarrow := \circlearrowleft^{-1}$

$\text{Can mkLft} = \{\Leftarrow\}$

Canonical elements

$\text{Can } c := \{(c\ a) * a \mid a \in \underline{A}\}$

Core functions $f : \text{Can } c \rightarrow \underline{B}$ capture the “actual” construction

$\text{merge}_{\leftarrow}^{\text{row}} : \text{Row}_n \rightarrow \text{Row}_n$

$$\begin{aligned}\text{merge}_{\leftarrow}^{\text{row}} \langle \rangle &:= \langle \rangle \\ \text{merge}_{\leftarrow}^{\text{row}} \left(\langle 2^k, 2^k \rangle \uparrow\uparrow r \right) &:= \langle 2^{k+1} \rangle \uparrow\uparrow \text{merge}_{\leftarrow}^{\text{row}}(r) \uparrow\uparrow \langle 0 \rangle \\ \text{merge}_{\leftarrow}^{\text{row}} (\langle 0 \rangle \uparrow\uparrow r) &:= \text{merge}_{\leftarrow}^{\text{row}}(r) \uparrow\uparrow \langle 0 \rangle \\ \text{merge}_{\leftarrow}^{\text{row}} (\langle x \rangle \uparrow\uparrow r) &:= \langle x \rangle \uparrow\uparrow \text{merge}_{\leftarrow}^{\text{row}}(r)\end{aligned}$$

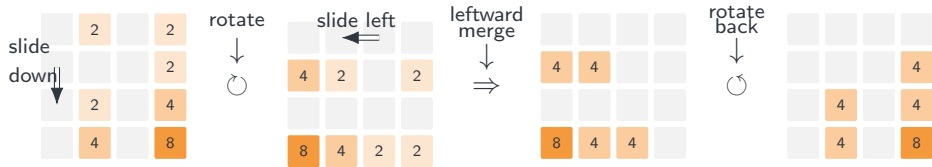
$\text{merge}_{\leftarrow} : \text{Board}_n \rightarrow \text{Board}_n$

$$\text{merge}_{\leftarrow} := \text{map } \text{merge}_{\leftarrow}^{\text{row}}$$

The **w.l.o.g. construction** captures the whole construction

Let A, B be G -actions, $c : \underline{A} \rightarrow \underline{G}$ a canoniser, and $f : \text{Can } c \rightarrow \underline{B}$ a core function.
Then

$$\text{wlog}(c, f) := \left(a \mapsto (ca)^{-1} \circledast f((ca) \circledast a) \right) : \underline{A} \rightarrow \underline{B}$$



TwentyFortyEight.hs

Indexed actions generalise actions to dependent types

Indexed action

Given a G -action Γ , an **indexed G -action** $\Gamma \vdash A$ consists of a **carrier** family $\underline{\Gamma} \vdash \underline{A}$; and a dependent function

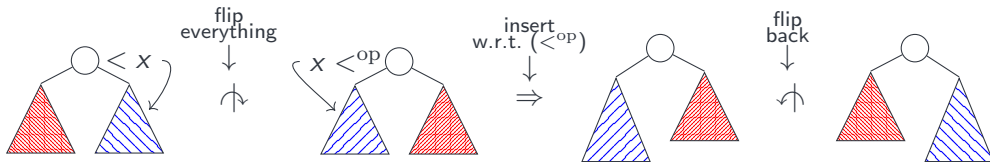
$$(\circledast) : (\underline{g} : \underline{G}) \rightarrow (\underline{\gamma} : \underline{\Gamma}) \rightarrow \underline{A}_{\underline{\gamma}} \rightarrow \underline{A}_{\underline{g} \circledast \underline{\gamma}},$$

such that

$$e_{\underline{\gamma}} \circledast a = a \qquad (g * h)_{\underline{\gamma}} \circledast a = g_{h \circledast \underline{\gamma}} \circledast (h_{\underline{\gamma}} \circledast a)$$

Proposition

A Γ -indexed action A amounts to an action $\coprod_{\Gamma} A$ over the Grothendieck construction for which the projection is equivariant: $\coprod_{\Gamma} A = \coprod_{\underline{\Gamma}} \underline{A}$ and $\pi_1 : \coprod_{\Gamma} A \circledast \rightarrow \Gamma$.



Tree.lean

Algebraic datatypes

A ***G*-symmetry strength** describes the action on constructors

Definition

Let G be a group and $F : \mathbf{Set} \rightarrow \mathbf{Set}$ a functor. A ***G*-symmetry strength** is a natural transformation

$$\mathrm{st}_X : \underline{G} \times FX \rightarrow F(\underline{G} \times X)$$

satisfying, for all $t \in FX$, $g, h \in \underline{G}$:

$$\mathrm{st}_X(e, t) = \mathrm{map}(x \mapsto \langle e, x \rangle) t$$

$$\mathrm{st}_X(g * h, t) = \mathrm{map}(\langle g', h', x \rangle \mapsto \langle g' * h', x \rangle) (\mathrm{st}_{\underline{G} \times X}(g, \mathrm{st}_X(h, t)))$$

A G -strength **lifts** G -actions from A to $F\underline{A}$

Proposition

For every symmetry strength st and every G -action A , there is a G -action $F_{\text{st}}A$ with $\underline{F_{\text{st}}A} := F\underline{A}$ and

$$\binom{F_{\text{st}}A}{*} : \underline{G} \times F\underline{A} \xrightarrow{\text{st}} F(\underline{G} \times \underline{A}) \xrightarrow{\text{map}(\binom{A}{*})} F\underline{A}$$

Moreover, if $h : A \circ \rightarrow B$ is equivariant then so is $\text{map } h : F_{\text{st}}A \circ \rightarrow F_{\text{st}}B$.

Two S_2 -strengths on $\text{Tree } \mathbb{N}$ induce two different actions

Flipping the tree

$$\text{st}^{\text{Tree } \mathbb{N}; S_2} \langle F, \text{Node } \ell n r \rangle := \text{Node } \langle F, r \rangle n \langle F, \ell \rangle$$

$$F \circledast \left(\begin{array}{c} & & 3 \\ & \swarrow & \searrow \\ 1 & 2 & 4 \end{array} \right) = \left(\begin{array}{c} 3 & & \\ \swarrow & & \searrow \\ 4 & 2 & 1 \end{array} \right)$$

Flipping the top node

$$\text{st}'^{\text{Tree } \mathbb{N}; S_2} \langle F, \text{Node } \ell n r \rangle := \text{Node } \langle I, r \rangle n \langle I, \ell \rangle$$

$$F \circledast' \left(\begin{array}{c} & & 3 \\ & \swarrow & \searrow \\ 1 & 2 & 4 \end{array} \right) = \left(\begin{array}{c} 3 & & \\ \swarrow & & \searrow \\ 1 & 2 & 4 \end{array} \right)$$

An equivariant algebra is **both an F -algebra and equivariant**

Definition

An **equivariant $\langle F, \text{st} \rangle$ -algebra** $\langle A, f \rangle$ consists of a G -action A and an equivariant F -algebra $f : F_{\text{st}} A \circledast \rightarrow A$, i.e., an F -algebra $f : F\underline{A} \rightarrow \underline{A}$ such that

$$f(g \overset{F_{\text{st}} A}{\circledast} t) = g \overset{A}{\circledast} f(t)$$

An **equivariant algebra homomorphism** $h : \langle A, f \rangle \circledast \rightarrow \langle B, g \rangle$ is simultaneously an F -homomorphism and equivariant $h : A \circledast \rightarrow B$.

The initial F -algebra is the **initial equivariant algebra**

For every F -algebra $f : FX \rightarrow X$ there is a unique **parameterised homomorphism** **pfold** $f : \underline{G} \times \mu F \rightarrow X$ satisfying

$$\mathbf{pfold} f \langle g, \mathbf{roll} t \rangle = f(\mathbf{map}(\mathbf{st}(g, t)))$$

Theorem

Let $\mathbf{roll} : F\mu F \rightarrow \mu F$ be an initial algebra for F with G -strength $\mathbf{st}$. Defining

$$(\circledast) := \mathbf{pfold} \mathbf{roll} : \underline{G} \times \mu F \rightarrow \mu F$$

makes $\langle \mu F, \mathbf{roll} \rangle$ into the **initial** equivariant $\langle F, \mathbf{st} \rangle$ -algebra: for every equivariant algebra $f : F_{\mathbf{st}} A \circledast \rightarrow A$, the unique F -homomorphism **fold** $f : \mu F \circledast \rightarrow A$ is equivariant.

Guaranteed equivariance

We want core functions to be **equivariant where defined**

Assume an equivariant $\varphi : A \circledast \rightarrow B$ extends a core function $f : \text{Can } c \rightarrow \underline{B}$. Then:

$$\begin{array}{ccccc} \varphi \text{ extends } f & \varphi \text{ is equivariant} & \varphi \text{ extends } f & & \\ \downarrow & \downarrow & \downarrow & & \\ f(g \circledast x) = \varphi(g \circledast x) & = g \circledast \varphi(x) & = g \circledast fx & & \end{array}$$

We want core functions to be **equivariant** where defined

Assume an equivariant $\varphi : A \circledast \rightarrow B$ extends a core function $f : \text{Can } c \rightarrow \underline{B}$. Then:

$$\begin{array}{ccccc} \varphi \text{ extends } f & \varphi \text{ is equivariant} & \varphi \text{ extends } f & & \\ \downarrow & \downarrow & \downarrow & & \\ f(g \circledast x) & = \varphi(g \circledast x) & = g \circledast \varphi(x) & = g \circledast fx \end{array}$$

Let A, B be G -actions and $X \subseteq \underline{A}$. Then, $f : X \rightarrow \underline{B}$ is **equivariant** when

$$\forall g \in \underline{G}. \forall x \in X. (g \overset{A}{\circledast} x \in X \implies f(g \overset{A}{\circledast} x) = g \overset{B}{\circledast} fx)$$

We want core functions to be **equivariant where defined**

Assume an equivariant $\varphi : A \circledast \rightarrow B$ extends a core function $f : \text{Can } c \rightarrow \underline{B}$. Then:

$$\begin{array}{ccccc} \varphi \text{ extends } f & \varphi \text{ is equivariant} & \varphi \text{ extends } f & & \\ \downarrow & \downarrow & \downarrow & & \\ f(g \circledast x) & = \varphi(g \circledast x) & = g \circledast \varphi(x) & = g \circledast fx \end{array}$$

Let A, B be G -actions and $X \subseteq \underline{A}$. Then, $f : X \rightarrow \underline{B}$ is **equivariant** when

$$\forall g \in \underline{G}. \forall x \in X. (g \overset{A}{\circledast} x \in X \implies f(g \overset{A}{\circledast} x) = g \overset{B}{\circledast} fx)$$

Theorem

Let A, B be G -actions, $c : \underline{A} \rightarrow \underline{G}$ a canoniser, $f : \text{Can } c \rightarrow \underline{B}$ an equivariant core function. Then $\text{wlog}(c, f) : A \circledast \rightarrow B$ is the unique equivariant function extending f .

Are canonisers **equivariant**?

$$(ca * g^{-1}) \circledast (g \circledast a) = ca \circledast a \in \text{Can } c$$

Are canonisers **equivariant**?

$$(ca * g^{-1}) \circledast (g \circledast a) = ca \circledast a \in \text{Can } c$$

Given a group G , we define the G -action G^{-1} with $\underline{G^{-1}} = \underline{G}$ and

$$g \overset{G^{-1}}{\circledast} h := h * g^{-1}$$

Are canonisers **equivariant**?

$$(ca * g^{-1}) \circledast (g \circledast a) = ca \circledast a \in \text{Can } c$$

Given a group G , we define the G -action G^{-1} with $\underline{G^{-1}} = \underline{G}$ and

$$g \overset{G^{-1}}{\circledast} h := h * g^{-1}$$

A canoniser $c : \underline{A} \rightarrow \underline{G}$ is **strictly equivariant** when

$$c(g \circledast a) = c a * g^{-1} = g \overset{G^{-1}}{\circledast} (c a)$$

i.e. when $c : A \circledast \rightarrow G^{-1}$.

Strict equivariance is a **strong assumption** and a **strong restriction**

Theorem

A canoniser $c : \underline{A} \rightarrow \underline{G}$ is strictly equivariant if and only if, for every G -action B , every core function $f : \text{Can } c \rightarrow \underline{B}$ is equivariant.

Strict equivariance is a **strong assumption** and a **strong restriction**

Theorem

A canoniser $c : \underline{A} \rightarrow \underline{G}$ is strictly equivariant if and only if, for every G -action B , every core function $f : \text{Can } c \rightarrow \underline{B}$ is equivariant.

Proposition

*If A has a strictly equivariant canoniser then A is **semi-regular**:*

$$g \circledast a = a \implies g = e$$

Example

C_4 -action on Board_n is not semi-regular, but it is on $(\text{Dir} \times \text{Board}_n)$. □

The issue is in **indistinguishability** of symmetries

Definition

For a G -action A and $a \in \underline{A}$, the **indistinguishability** relation is

$$g \overset{a}{\sim} h \iff g \circledast a = h \circledast a$$

The quotient $(G/A)_a := G / \overset{a}{\sim}$ carries an A -indexed G -action:

$$g \circledast_a [h]_{\overset{a}{\sim}} := [h * g^{-1}]_{g \circledast_a}$$

Side note: **orbit-stabiliser theorem** is an **equivariant isomorphism**

$$\begin{aligned} G \circledast [a] &:= \{ h \circledast a \mid h \in \underline{G} \} \\ g \circledast_a (h \circledast a) &:= h \circledast a = (h * g^{-1}) \circledast (g \circledast a) \in G \circledast [g \circledast a] \end{aligned}$$

Theorem (Orbit-Stabiliser)

Let A be a G -action. There is an equivariant isomorphism:

$$\varphi_a : (G/A)_a \xrightarrow[\circledast]{\cong} G \circledast [a] \quad \varphi_a[h]_{\sim a} := h \circledast a$$

Collapsing indistinguishable symmetries **weakens** strict equivariance

A canoniser $c : \underline{A} \rightarrow \underline{G}$ is **equivariant** when $(a \mapsto [c\ a]_a) : A \circlearrowright G/A$.

Example

$F \circledast \langle a, a \rangle = \langle a, a \rangle$, so $I \overset{\langle a, a \rangle}{\sim} F$, thus

$$c(F \circledast \langle a, a \rangle) = I \overset{\langle a, a \rangle}{\sim} F = F \underset{\langle a, a \rangle}{\circledast} c \langle a, a \rangle$$



Theorem

Using the axiom of choice, every G -action A has an equivariant canoniser $c : \underline{A} \rightarrow \underline{G}$.

Equivariant canonisers make every **stable** core function equivariant

Every equivariant f satisfies $g \in G_a \implies g \circ f\ a = f(g \circ a) = f\ a$, so

$$G_a \subseteq G_{f\ a}$$

A

core function $f : \text{Can } c \rightarrow \underline{B}$ is **stable** when $\forall a \in \text{Can } c. G_a \subseteq G_{f\ a}$.

Theorem

c is equivariant iff every stable core function $f : \text{Can } c \rightarrow \underline{B}$ is equivariant.

Next time, on **WG 2.1...**

Next time, on **WG 2.1**...

... for you in the paper

- more examples (DAGs, BSTs)
- indexed datatypes
- Harrison's tactic (non)-examples
- EWD1223

... for us still in the works

- more examples (???)
- mechanisation
- programming abstractions
- efficient implementation